Analytical models for performance and energy consumption evaluation of storage devices

Eric Borba, Eduardo Tavares, Paulo Maciel, and Carlos Gomes
Centro de Informática, *Universidade Federal de Pernambuco*, Recife, Brasil
{erb,eagt,prmm,cga}@cin.ufpe.br

***Abstract*—Improvements in data storage may be constrained by the lower performance of hard disk drives (HDD) and the higher cost per gigabyte of solid-state drives (SSD). To mitigate these issues, hybrid storage architectures have been conceived. Some works evaluate the performance of storage architectures, but energy consumption is usually neglected and not simultaneously evaluated with performance. This paper presents an approach based on generalized stochastic Petri nets (GSPN) for performance and energy consumption evaluation of individual and hybrid storage systems. The proposed models can represent distinct workloads and also estimate throughput, response time and energy consumption of storage systems. Some case studies based on industry-standard benchmarks are adopted to demonstrate the feasibility of the proposed approach.**

***Index Terms*—performance evaluation, stochastic Petri nets, data management, energy consumption, hybrid storage, cloud computing**

I. Introduction

Energy consumption in data centers is a critical and challenging issue, which has motivated many studies to reduce operational costs. For instance, reports indicate the cost of energy consumed by a server (during its lifetime) will exceed the hardware costs, if current demand continues to increase further [1].

Cloud computing has been widely adopted for representative companies and institutions, since this paradigm reduces operational costs and improves the utilization of computational resources. For instance, the United States Library of Congress has moved its digital content to a cloud storage provider, and Netflix adopts Amazon S3 platform for storing its videos [2]

Nevertheless, energy consumption of cloud computing systems also needs to be addressed, as the amount of stored data and applications using this paradigm steadily increases [3]. 90% of all data in the world was generated over the last 2 years [4], and global data are predicted to reach 163 zettabytes by 2025 [5]. Facebook and Google respectively process 20 [6] and 25 petabytes of data per day [7].

Storage subsystems are a major part of cloud computing, as they contemplate 35% of cloud computing costs [3] and are responsible for more than 27% of energy consumption. Besides, storage devices may be a bottleneck for computer systems [8], as they may take about 90% of a transaction execution time [9].

Regarding the technology for storage devices, solid-state drives provide faster read operations than magnetic hard disks [10]. However, for some workloads, SSDs may not provide better results than HDDs regarding sequential access. Alternatively, hybrid approaches have been proposed. Hybrid storage systems may have higher performance than HDD storage with affordable cost, becoming a very promising solution for many systems, such as those based on cloud computing [11].

As a consequence, researches concerning storage architectures have been carried out [12]. Although many works evaluate the performance of storage systems, energy consumption is usually neglected and not simultaneously evaluated with performance. In this context, performance models [13] are quite important, as different designs and architectures can be assessed before implementing the real system.

This paper proposes an approach based on generalized stochastic petri nets (GSPN) for performance and energy consumption evaluation of homogeneous (e.g., only HDD) and hybrid storage systems. The proposed models can represent distinct workloads, and they may also estimate throughput, energy consumption and response time. Case studies based on industry-standard benchmarks demonstrate the feasibility of the proposed approach.

II. Related Work

Hybrid storage is a prominent research field that has motivated many studies: architectures to integrate HDDs and SSDs; analytical models for performance estimate; and data placement techniques for putting specific data on the most suitable storage component.

In [14], the authors propose a mechanism for integrating SSD devices with HDD using a hybrid file system. However, that work only investigates individual drive performance and, based on the results, an ideal super block structure is presented without experimentation or simulation. Tan et al. [15] study the effectiveness of Hadoop Distributed File System (HDFS) on a SSD-HDD storage. Experiments utilize different architectures, which are submitted to three types of big data workloads. Although results indicate the benefits of adopting SSDs, the authors indicate a workload-aware architecture may obtain better results.

Mingzhou et al. [16] propose a numerical approach based on Markov chains to estimate the service time of HDDs, considering random accesses. The arrival of read or write requests is represented as a Poisson process. Experimental results are presented to validate their approach. In [17], the authors present an analytical model based on Markov decision process to estimate the hit ratio (i.e., number of cache hits to the number of lookups) of hybrid storages.

To improve energy consumption and reliability of hybrid storage systems, Jingyu et al. [18] suggest storing metadata in SSDs separated from data files. Results show 70% of energy savings. Boukhelef et al. [11] propose a hybrid storage system to deal with data placement problem. Optimization algorithms are also presented to better place data based on user requirements. Results indicate performance can be improved up to 40%.

Different from previous works, this paper proposes models based on GSPN for evaluating the performance and energy consumption of homogenous and hybrid storage systems. Additionally, the proposed models provide a graphical representation of workload features. This work also takes into account real-world workloads to demonstrate the practical suitability of the conceived models.

III. Generalized stochastic Petri nets

Petri nets (PN) [19] are a family of formalisms very well suited for modeling several system types, since concurrency, synchronization, communication mechanisms as well as deterministic and probabilistic delays are naturally represented. In general, a Petri net is a bipartite directed graph, in which places (represented by circles) denote local states and transitions (depicted as rectangles) represent actions. Arcs (directed edges) connect places to transitions and vice-versa. Tokens (small filled circles) may reside in places, which denote the state (i.e., marking) of a PN.

This work adopts generalized stochastic Petri nets (GSPN) [19], which is a prominent PN extension that allows the association of exponential distribution to timed transitions (represented by white rectangles), or zero delays to immediate transitions (depicted as thin black rectangles). The state space of GSPN models may be translated into continuous-time Markov chains (CTMC) [20], and simulation techniques may also be adopted for estimating performance metrics, as an alternative to the Markov chain generation.

In GSPNs, non-exponential delays may be approximated using phase-type distributions [21], more specifically, Erlang, hyperexponential and hypoexponential. A trapezoidal transition, namely, s-transition, is adopted to denote a subnet, which models a delay using a phase-type distribution. Particularly, this paper adopts the phase approximation technique described in [21], which is an algorithm to match first and second central moments of a distribution.

GSPN is a suitable formalism for this work, as it contemplates a graphical representation of actions with zero (e.g., to represent workload features) and non-exponential delays. More specifically, such a formalism reduces the complexity of modeling (without loss of reliability on results) and explaining the proposed approach. For instance, different from queueing network models, GSPNs can model features such as blocking, synchronization, priority queuing disciplines, and operations to acquire and hold multiple resources [21].

IV. Performance modeling

This section presents the performance models conceived for representing storage systems. The models allow the representation of read and write operations under different workloads, access patterns (random or sequential) and object sizes. Besides, we have conceived our modeling approach for stationary analysis [19], in which (without loss of generality) the analysis assumes a system's long run.

Two models are proposed and they are based on GSPN formalism: (i) single storage model; and (ii) multiple storage model. The single storage model represents client requests to a system with a single storage device (e.g., HDD) or a hybrid system as a black box (i.e., without distinguishing its components). The multiple storage model is adopted for assessing the impact of workloads on different arrangements of storages (i.e., hybrid storage systems). Unlike the single model, this approach allows system designers to explicitly evaluate the components of hybrid systems.

The metrics of interest are throughput, mean response time, and energy consumption. Throughput represents input/output per second (IOPS) [22], which estimates the amount of processed requests (write or read) in one second. Mean response time is the average time for a single operation to complete. To estimate energy consumption, the same parameters (i.e., the proportion of each workload feature) of models are considered.

For the sake of explanation, we present the multiple storage model with only two different devices (HDD and SSD). However, this is not a limitation of the model, which is capable of representing storage systems with additional components (e.g., 4 HDDs; 2 SDDs and 4 HDDs). Additional storages may lead to state space size explosion [23], but simulation techniques may also be taken into account, as an alternative to CTMC generation [24].

Besides, specific features, such as metadata manipulation, are not explicitly represented on the conceived models, as, in the context of storage devices, there is no distinction of the data type being accessed or stored. Basic components are also abstracted since we have been concerned only with the whole storage device. As an example, for the sake of simplicity, cache memories are not represented explicitly. Similarly, filesystems and respective policies are not analyzed, as, in the context of this work, such a data structuring is not detailed. This abstraction level allows the assessment of different systems without dealing with a detailed model that may not be feasibly evaluated.

For characterizing workloads, we have taken into account some considerations due to the various possibilities of attributes. For the sake of tractability, we have adopted only two access patterns: random and sequential. Concerning object sizes, we have assumed a classification typically used in the literature to cover distinct operational characteristics found in storage devices [25] [26] [27]. In such a classification, SSDs and HDDs' performance distinguishes considerably for processing small and large objects. These considerations contemplate the most relevant standards for assessing storage

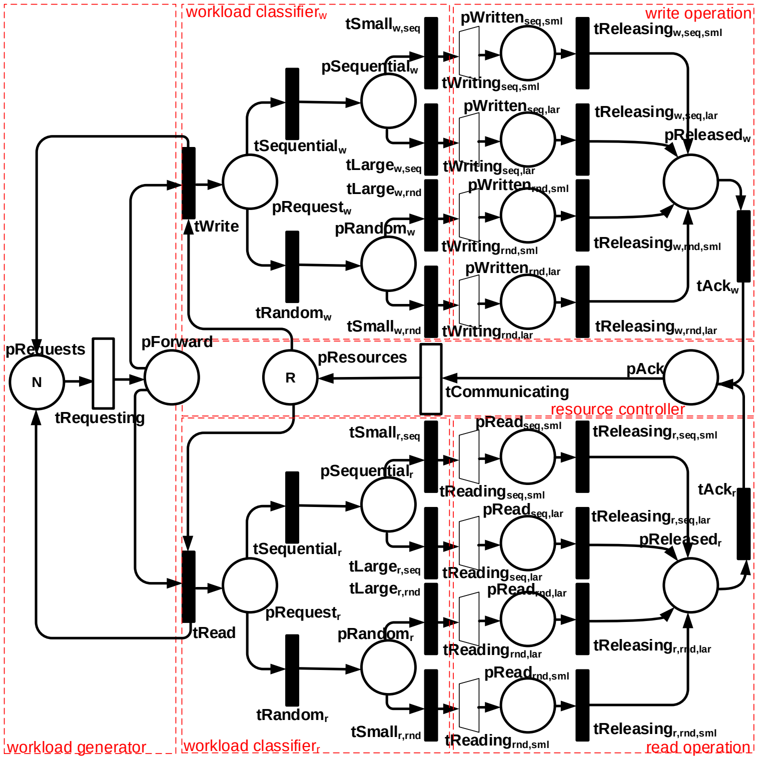


Fig. 1: Single storage model.

devices [28]. Nevertheless, this modeling approach does not contemplate the impacts of several request sizes on storage devices' performance (i.e., any request is classified only as small or large). Additionally, as a limitation, this approach has not contemplated features for evaluating issues as caching, write buffering, prefetching, and write amplification (the ratio of total intern writes to externally-requested writes).

The following notation is adopted: $E\{\#p\}$ represents the mean value of the inner expression, in which $\#p$ denotes the number of tokens in place p; and $W(T)$ represents the firing rate associated with transition T.

Additionally, function $\eta : T_{imm} \to [0,1]$ maps each immediate transition ($t \in T_{imm}$) to a normalized weight. More specifically, the weights represent the transition firing probability in a conflict set [19], and, for the adopted models, each immediate transition can only be in one conflict set. Next sections present the models using building blocks (i.e., submodels).

A. Single storage model

Figure 1 depicts the GSPN model for representing systems with a single storage.

workload generator block is responsible for representing user requests. The marking of place $pRequests$ (N) represents the allowed number of concurrent requests in the system, and each token is a client (worker) request. Transition $tRequesting$ indicates the arrival of a request within a storage. This transition adopts *infinite server semantics* [19] in order to represent concurrent arrivals. Tokens in place $pForward$ represent the request prepared for writing ($tWrite$) or reading ($tRead$).

A block *workload classifier$_{op}$* is adopted for each operation. Transitions $tWrite$ and $tRead$ denote the amount of requests for the respective activity, and they have weights indicating the probability of each operation [29]. For instance, in $mixed$ operations, read and write may have the same probability (0.5). Tokens in places $pRequest_{op}$ indicate read or write requests are queued. Immediate transitions $tSequential_{op}$ and $tRandom_{op}$ define the access pattern for a workload, and, similarly, their weights indicate the amount of requests associated with each pattern. Transition $tSmall_{op,pt}$ and $tLarge_{op,pt}$ represent the object size.

write and *read operation* blocks model the operation execution, and the delay is denoted by s-transition $tWriting_{pt,os}$ and $tReading_{pt,os}$. Tokens in places $pWritten_{pt,os}$ and $pRead_{pt,os}$ represent the conclusion of an activity. $tReleasing_{op,pt,os}$ and $tAck_{op}$ indicate the notification of resource release to the storage controller.

resource controller block denotes the storage readiness to execute read or write operations. A token in place $pAck$ indicates a resource is ready to be released, in which the communication with the controller is depicted by transition $tCommunicating$. Besides the marking of place $pResource$ (R) indicates the storage is ready for executing one or more operations.

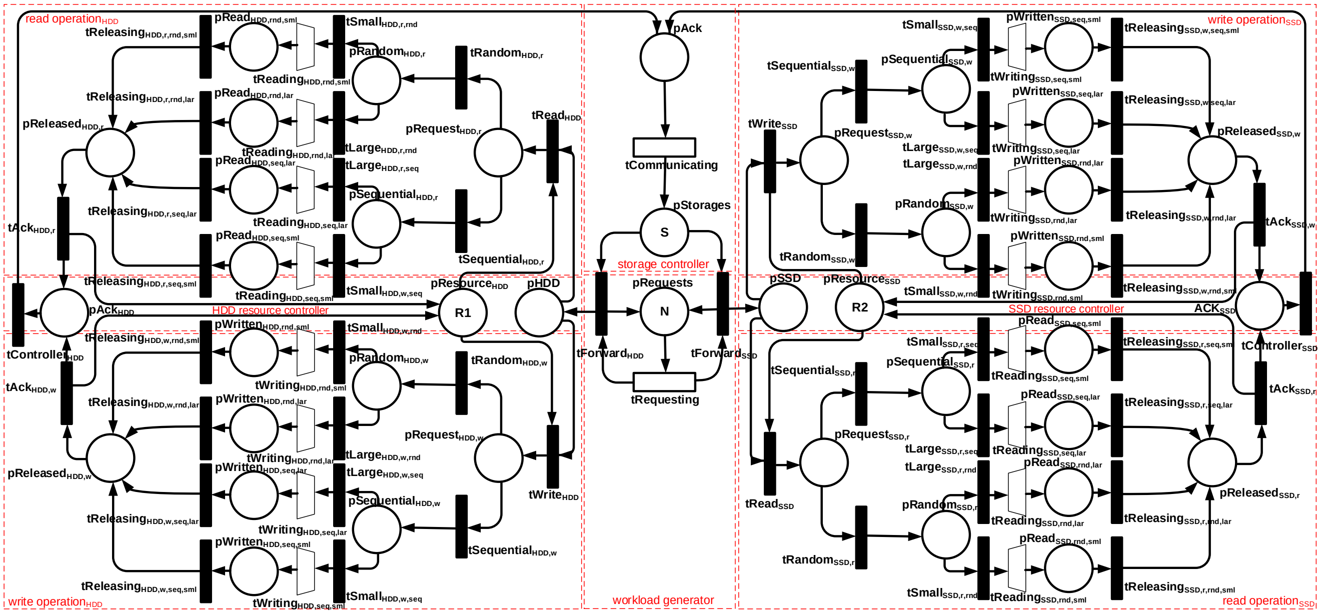


Fig. 2: Multiple storage model.

For the proposed model, mean response time is estimated using Little's law [20] $R = L/\lambda$, in which R represents the mean response time, L is the average number of requisitions and λ represents the arrival rate of requests. For this model, $L = N - E\{\#pRequests\}$ and $\lambda = E\{\#pRequests\} \times W(tRequesting)$. System throughput (i.e., IOPS) is estimated as $TH = E\{\#pAck\} \times W(tCommunicating)$.

For energy consumption, the workload features (e.g., access pattern) must be taken into account, as they influence the system power consumption. This work takes into account the proportion of each factor, which is represented as weights in immediate transitions ($\eta(t)$). For the single device model, the following weights are taken into account: $\eta(tWrite) = \kappa$; $\eta(tRead) = 1 - \kappa$; $\eta(tRandom) = \alpha$; $\eta(tSequential) = 1 - \alpha$; $\eta(tSmall) = \beta$; and $\eta(tLarge) = 1 - \beta$. System energy consumption (EC) is then estimated as follows

$$\begin{aligned} EP_w =& \kappa \left(EP_{w_1} * \alpha * \beta + EP_{w_2} * (1 - \alpha) * \beta + EP_{w_3} \right. \\ & * \alpha * (1 - \beta) + EP_{w_4} * (1 - \alpha) * (1 - \beta) \end{aligned} \tag{1}$$

$$\begin{aligned} EP_r =& (1 - \kappa) * (EP_{r_5} * \alpha * \beta + EP_{r_6} * (1 - \alpha) * \beta + \\ & EP_{r_7} * \alpha * (1 - \beta) + EP_{r_8} * (1 - \alpha) * (1 - \beta) \end{aligned} \tag{2}$$

$$EC = (EP_w + EP_r) * TH * time \tag{3}$$

EP_{op} is the mean power consumption for an operation (read - r or write - w), which is estimated using the mean power of each workload feature. For instance EP_{w1} denotes the power of a write operation (w) using random access (α) and an small object (β). $time$ is the time of interest.

B. Multiple storage model

Figure 2 depicts the GSPN model for representing systems with multiple storage devices. For a better understanding, this section presents the model using a hybrid storage system (1 SSD and 1 HDD).

Similar to previous model, *workload generator* block represents the creation of user requests, in which the marking N in place $pRequests$ indicates the number of concurrent requests. Timed transition $tRequesting$ adopts *infinite server semantic* to represent concurrent arrivals. Immediate transitions $tForward_d$ denote a request is redirected to a storage d. Tokens in places $pHDD$ and $pSSD$ ($pStorage$) indicate read or write requests are queued in a storage device.

Similar to single storage model, *read operation$_d$* and *write operation$_d$* blocks represent, respectively, the reading and writing activities. For each storage device in the system, both blocks are adopted.

resource controller$_d$ block models the available resources for performing an operation in a request. The number of tokens (e.g., R_1) in places $pResource_d$ denotes the number of operations are concurrently carried out. Transition $tController_d$ represent the device is informing the controller about the conclusion of an operation.

In *storage controller* block, a token in place $pAck$ represents a storage concluded the operation, and transition $tCommunicating$ denotes the controller delay for receiving the acknowledgment. This work assumes the storage controller can simultaneously receive acknowledgments from all devices (i.e.,*infinite server semantics*). The marking in place $pStorages$ (S) denotes the number of devices in the system.

Mean response time is estimated as $R_h = (N - E\{\#pRequests\})/(E\{\#pRequests\} \times$

$W(tRequesting))$. Throughput is $TH_h = E\{\#pAck\} \times 1/W(tCommunicating)$. Energy consumption ($EC_h$) is obtained from the power consumption of the workload features ($EP_{d,op,i,j}$) in all storage devices (n):

$$EC_h = \left(\sum_{d=0}^{n} \eta(tForward_d) * EP_d \right) * TH_h * time \quad (4)$$

$$EP_d = \sum_{op} \sum_{i} \sum_{j} \eta(op) * \eta(i) * \eta(j) * EP_{d,op,i,j} \quad (5)$$

in which $op \in (tWrite_d, tRead_d)$, $i \in (tSequential_{d,op}, tRandom_{d,op})$ and $j \in (tSmall_{d,op,i}, tLarge_{d,op,i})$

The model has been presented considering two distinct devices for a hybrid system. However, additional devices can be included by considering additional $read$, $write$ and $resource\ controller$ blocks.

V. Design of experiments

This work adopts an approach based on design of experiments (DoE) [30] for evaluating storage systems. More specifically, a factorial design (i.e., all possible combinations of the levels of the factors are investigated) is adopted ($\prod_{i=1}^{k} l_i$), and four experiments are carried out using GSPN models.

The first experiment adopts a screening approach for identifying the magnitude of each factor and interactions. Five factors ($k = 5$) are taken into account: (i) storage technology ($technology$); (ii) object size ($object\_size$); (iii) operation type ($operation$); (iv) access pattern ($pattern$); and (v) number of threads ($workers$). Table I depicts the levels (l_i) for each factor, and the metrics of interest are response time, IOPS and energy consumption. For the sake of validation and comparison in the experiments, we assume the energy consumption for one second.

Two additional experiments are adopted, which utilize the results from the screening approach and the guidelines for benchmarks developed by the Storage Performance Council (SPC) [31]. Such a council is composed of representative companies that define methodologies to evaluate storage devices and systems. The experiments also utilize two supplementary metrics: (i) IOPS/energy consumption, and (ii) price/IOPS. The former represents energy efficiency and a higher value is better. The latter is the relation between storage system price and performance, and lower values are preferable. Storage system price is calculated as storage capacity × cost per GB. In this work, we have considered US$0.075 and US$1.0 [32] per gigabyte for HDD and SSD, respectively.

TABLE I: Screening - factors and levels.

factor	levels
technology	1TBHDD, 120GBSSD, Hybrid
object_size	4KB, 1MB
operation	write, read
workers	1, 4
pattern	random (rnd), sequential (seq)

TABLE II: Experiment components.

component	description
1TBHDD	HDD 1TB
120GBSSD	SSD 120GB
Hybrid	HDD 1TB + SSD 120GB
server	quad-core, 3.10GHz, 8GB RAM

Similar to the screening approach, the same five factors are considered, and, except for $technology$, we fixed the levels to represent specific workloads. The experiments are explained as follows.

The second experiment evaluates the performance of storage systems, in which the application's access pattern is predominately random (e.g., database systems). In this case, the objects are stored on device blocks without a specifc order [33]. Besides, write operations contemplate 70% of the workload ($70\%\_w$). This experiment, namely, *random access*, contemplates the following factors and levels: (i) $technology$ - $1TBHDD$, $120GBSSD$ and $Hybrid$; (ii) $object\_size$ - $4KB$; (iii) $operation$ - $70\%\_w$; (iv) $pattern$ - rnd; and (v) $workers$ - 4.

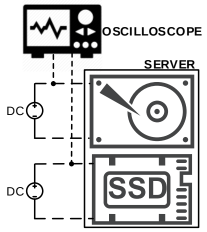


Fig. 3: Environment setting

The third experiment (*sequential access*) assesses the behavior of storage systems for applications that require large-scale sequential data access (e.g., financial processing). Sequential access assumes the objects are stored on contiguous blocks in the storage devices [34]. The workload also assumes equal proportion (50%) of write and read operations ($50\%\_w$). The experiment considers the following levels: (i) $technology$ - $1TBHDD$, $120GBSSD$ and $Hybrid$; (ii) $object\_size$ - $1MB$; (iii) $operation$ - $50\%\_w$; (iv) $pattern$ - seq; and (v) $workers$ - 4.

The fourth experiment, namely, *mixed*, represents raw data workloads, which usually are composed of small random requisitions (80% - $80\%\_rnd$) and commonly have mixed operations (50% write - $50\%\_w$) from simultaneous clients (e.g., 4 workers) [31] [35]. The workload also assumes 20% of sequential requisitions with large object sizes (1MB) ($20\%\_los$). This experiment takes into account the following levels: (i) $technology$ - $1TBHDD$, $120GBSSD$ and $Hybrid$; (ii) $object\_size$ - $20\%\_los$; (iii) $operation$ - $50\%\_w$; (iv) $pattern$ - $80\%\_rnd$; and (v) $workers$ - 4.

TABLE III: Moment matching - HDD and SSD

			1TBHDD				**120GBSSD**			
op	**os**	**pt**	**mean (ms)**	**st.dev. (ms)**	**phases**	**distribution**	**mean (ms)**	**st.dev. (ms)**	**phases**	**distribution**
write	4KB	rnd	3.510000	0.950510	10	hypo.	0.968840	0.778870	1	hypo.
		seq	0.072336	0.024602	8	hypo.	0.223670	0.142030	2	hypo.
	1MB	rnd	9.920000	3.820000	6	hypo.	29.950000	17.410000	2	hypo.
		seq	5.690000	0.131970	10	hypo.	9.930000	4.540000	4	hypo.
read	4KB	rnd	8.000000	0.839160	10	hypo.	0.612730	0.056047	10	hypo.
		seq	0.047958	0.019129	6	hypo.	0.210620	0.040588	10	hypo.
	1MB	rnd	14.190000	0.276980	10	hypo.	4.470000	0.234440	10	hypo.
		seq	5.580000	0.107310	10	hypo.	4.010000	0.022829	10	hypo.

A. Tools

This work adopts Iometer tool [36] [37] [38] to characterize storage devices for read and write operations. The results are utilized on the conceived GSPN models for validation and experiments. We utilize the same factors and levels in Table I, but only one $worker$ is considered.

Figure 3 depicts the adopted system, whose components are detailed in Table II. Using Iometer, the server performs workload on each drive (or simultaneously for the $hybrid$ approach). Simultaneously, an oscilloscope collects instantaneous voltage (using shunt resistors), the power is estimated and, then, energy consumption is obtained using numerical integration. For each treatment, the system collects 20 samples to estimate the mean delays for the read/write operations and the metrics of interest (i.e., IOPS, mean response time and energy consumption).

This work adopts Mercury [39] and TimeNET [40] tools for evaluating GSPN models. The validation has been carried out on a Intel core 2 Duo 2.4GHz, 8GB RAM, Windows 10. The average time for evaluating each model (i.e., Markov chain generation and numerical solution) is around 5s.

VI. Experimental results

This section presents the validation for the conceived GSPN models and experimentals results to demonstrate the practical feasibility of the proposed approach.

A. Validation

This work demonstrates the validation for the single storage model, since this approach also provides insights to the multiple storage model.

We have performed experiments with real systems (Section V-A) and compared to the values obtained with GSPN models (using stationary analysis). The single model is validated for HDD and SSD storages. Besides, a single object size is utilized ($4KB$) and two access patterns (pt) are assumed: random (rnd) and sequential (seq). The validation also takes into account read and write operations (op).

The models utilize a delay of 1μs (following an exponential distribution) for transition $tRequesting$. For all GSPN models, the marking of place $pRequests$ is 1, which denotes only one $worker$. The marking Rx (place $pResources_d$) is also 1, as we have adopted storages ($1TBHDD$ and $120GBSSD$) with serial advanced technology attachment (SATA) and advanced host controller interface (AHCI). In these technologies, a device do not carry out simultaneous operations.

The delays for write and read operation have been approximated using phase-type distributions. Table III details the results for the moment matching, considering data collected on the real system using Iometer. $mean$ is the mean delay and $st.dev.$ is the standard deviation. $distribution$ denotes the probability distribution, and hypoexponential ($hypo$) has been adopted [21]. $phases$ represents the number of phases. We adopt a limit of 10 phases, since additional phases do not influence the results [41]. Table IV shows the mean power of each drive ($1TBHDD$ and $120GBSSD$) for distinct workload features.

Table V depicts the values for the real systems and the estimates using the single storage model. The metrics are energy consumption, response time and IOPS^{-1}. For all metrics, the model values are contained in the 95% confidence intervals (95% $c.i.$) obtained from the systems and, thus, the hypothesis of equivalence cannot be refuted.

TABLE IV: Mean power values.

			power (W)	
op	**os**	**pt**	**1TBHDD**	**120GBSSD**
write	4KB	rnd	0.0102094	0.0008643
		seq	0.0003109	0.0001321
	1MB	rnd	0.0411532	0.0328646
		seq	0.0229215	0.0075362
read	4KB	rnd	0.0252030	0.0004737
		seq	0.0001598	0.0001919
	1MB	rnd	0.0455828	0.0086180
		seq	0.0183449	0.0081872

B. Experiment I: Screening

This experiment assesses the effects of each factor and interactions based on DoE detailed in Section V. Effect is the change in response due to a change in the factor level, and Table VI shows a rank for main and second-order interactions. The rank is ordered in descending order taking into account the absolute values of all effects.

This work considers only main effects and second-order interactions, since high order interactions do not considerably impact the adopted metrics [30]. Besides, the nine most significant effects are illustrated, as other effects do not remarkably affect the metrics. For $technology$ factor, the adopted levels

TABLE V: Validation results - single storage model.

			energy consumption (J)		response time (ms)		IOPS^{-1}	
device	op	pt	95% c.i.	GSPN	95% c.i.	GSPN	95% c.i.	GSPN
1TBHDD	write	rnd	(2.8450; 2.9597)	2.8909	(3.4862; 3.5218)	3.5190	(0.003400; 0.003523)	0.003520
		seq	(3.5052; 4.3209)	3.9166	(0.0808; 0.0823)	0.0813	(0.000081; 0.000083)	0.000082
	read	rnd	(2.9513; 3.1926)	3.0812	(8.0046; 8.0584)	8.0090	(0.008007; 0.008061)	0.008010
		seq	(2.8750; 2.9931)	2.9007	(0.0561; 0.0574)	0.0569	(0.000056; 0.000057)	0.000057
120GBSSD	write	rnd	(0.8625; 0.9753)	0.8830	(0.7839; 1.0923)	0.9778	(0.000785; 0.001095)	0.000978
		seq	(0.7909; 0.8507)	0.8195	(0.1456; 0.1754)	0.1602	(0.000146; 0.000176)	0.000161
	read	rnd	(0.7598; 0.7708)	0.7647	(0.6145; 0.6217)	0.6217	(0.000615; 0.000622)	0.000622
		seq	(0.8585; 0.8875)	0.8701	(0.2183; 0.2200)	0.2196	(0.000219; 0.000220)	0.000220

TABLE VI: Rank of main and interaction effects.

energy consumption (J)		response time (ms)		IOPS	
factor/interaction	effect	factor/interaction	effect	factor/interaction	effect
technology(120GBSSD-Hybrid)	5.6919	object_size	20.1874	object_size	721.7838
technology(1TBHDD-Hybrid)	3.6281	workers	16.2618	technology(1TBHDD-Hybrid)	285.0174
operation*technology(120GBSSD-Hybrid)	2.1027	pattern	9.7670	technology(120GBSSD-Hybrid)	281.6786
object_size*technology(1TBHDD-Hybrid)	2.0719	operation	7.2928	object_size*technology(120GBSSD-Hybrid)	270.4418
technology(1TBHDD-120GBSSD)	2.0637	technology(120GBSSD-Hybrid)	3.4529	pattern	218.5956
object_size	2.0182	technology(1TBHDD-Hybrid)	2.8260	operation*technology(120GBSSD-Hybrid)	188.1111
object_size*technology(120GBSSD-Hybrid)	1.8911	technology(1TBHDD-120GBSSD)	0.6269	operation*technology(1TBHDD-120GBSSD)	146.8808
object_size*operation	1.2833	object_size*technology(1TBHDD-120GBSSD)	0.0060	pattern*technology(1TBHDD-Hybrid)	144.0635
operation	1.0491	operation*technology(1TBHDD-Hybrid)	0.0040	object_size*operation	136.4913

TABLE VII: Experimental results.

experiment	technology	energy consumption (J)	response time (ms)	IOPS
random accesses	SSD	0.850	3.162	1264.690
	HDD	2.937	17.248	231.897
	Hybrid	6.179	8.636	926
sequential access	SSD	2.055	36.901	108.396
	HDD	3.580	21.299	187.793
	Hybrid	10.734	14.173	564.599
mixed	SSD	1.571	10.862	368.226
	HDD	3.089	19.711	202.922
	Hybrid	6.639	11.746	681.211

for estimating an effect are indicated in parenthesis (e.g., $technology(120GBSSD - Hybrid)$).

Considering energy consumption, $technology$, $object\_size$, $operation$ and respective interactions (e.g., $operation * technology(120GBSSD - Hybrid)$) are the most significant effects. Nevertheless, the adoption of a hybrid system (i.e., $technology(120GBSSD - Hybrid)$ and $technology(1TBHDD - Hybrid)$) considerably contribute to energy consumption (change of $5.6919J$ and $3.6281J$).

The main effects account for most of the impact on response time, and interactions do not affect significantly this metric. $object\_size$ and $workers$ are the factors with considerable variation: $20.1874ms$ and $16.2618ms$, respectively. $Hybrid$ is the best level for $technology$, since it reduces response time in 3.4529ms and 2.8260ms compared to $120GBSSD$ and $1TBHDD$.

Regarding IOPS, $object\_size$ has the greatest influence followed by $technology$. $Hybrid$ level considerably improves IOPS, as it may increase throughput more than 280 operations per second. $pattern$ also influences system throughput: rnd - 158.3573 and seq - 376.9530. Besides, some interactions also have an effect on IOPS, for instance, $object\_size * technology(120GBSSD - Hybrid)$ (270.4418), and $operation * technology(120GBSSD - Hybrid)$ (188.1111).

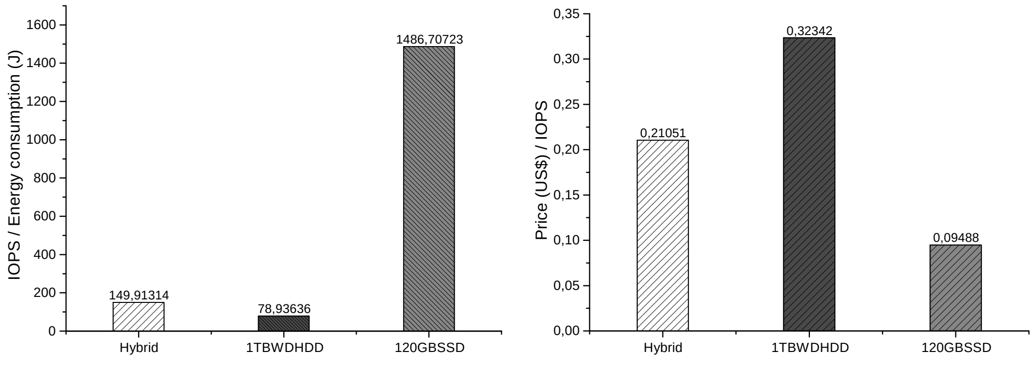
Results show the factors do not similarly influence all metrics (i.e., have the same rank position). Thus, for the next experiments, we fix and mix some factors levels to better assess the effects on storage systems. Henceforth, four $workers$ are adopted, since real-workloads are usually composed of concurrent requests [35].

C. Experiment II: Random access

This section presents results for storages considering a workload mainly composed of random requests (Table VII).

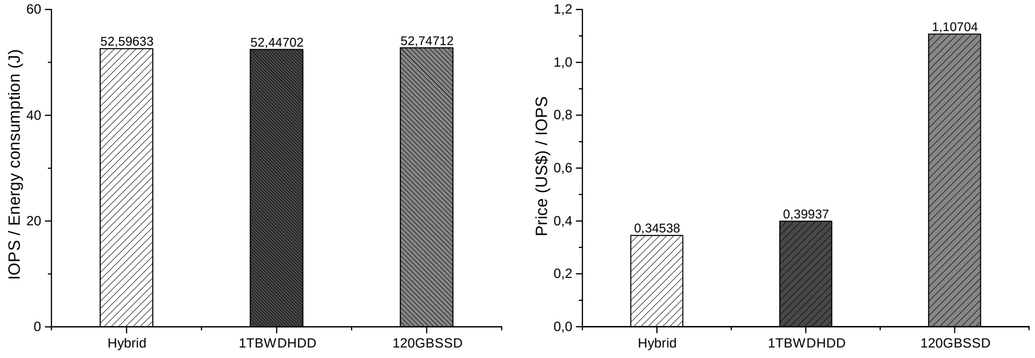
Results indicate $120GBSSD$ as the best technology regarding all metrics, due to the absence of mechanical components. The performance of magnetic disks is jeopardized because of excessive disk rotations. Compared to $1TBHDD$, $Hybrid$ has better values for response time and IOPS, but hybrid system consumes more energy. Considering the ratios IOPS/energy consumption and price/IOPS (Figures 4(a) and 4(b)), SSD has better results followed by hybrid system.

Usually, SSDs are known by the remarkable performance for read operations [42]. Additionally, this experiment corroborates the ability of SSDs to handle random requests, even under a workload consisting mostly of write requests (70%) [43].



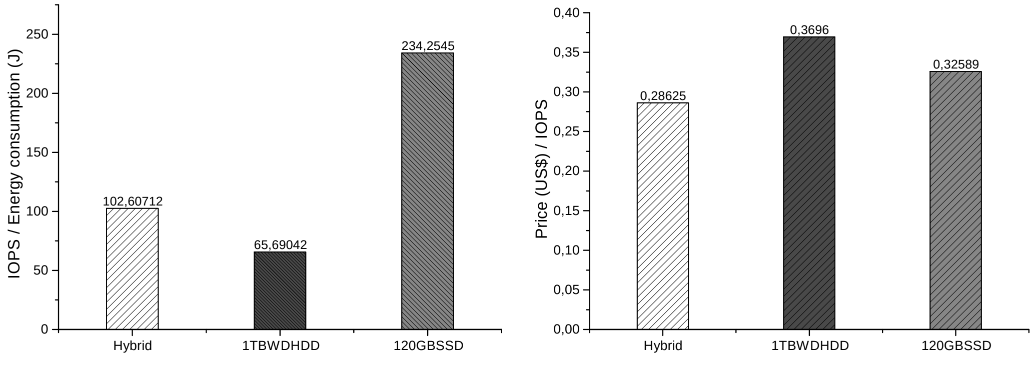
(a) (b)

Fig. 4: ***Random access*** **- (a)** IOPS/energy consumption**; and (b)** price/IOPS.



(a) (b)

Fig. 5: ***Sequential access*** **- (a)** IOPS/energy consumption**; and (b)** price/IOPS.



(a) (b)

Fig. 6: ***Mixed access*** **- (a)** IOPS/energy consumption**; and (b)** price/IOPS.

D. Experiment III: Sequential access

This section takes into account a workload represented by sequential requests.

Similar to previous experiment, the hybrid system has the worst values for energy consumption (Table VII). However, this system is capable of reducing response time (33.45%) and increasing IOPS (200.64%), comparing to $1TBHDD$ (commonly considered the most suitable technology for sequential workloads [44]). Results highlight the improvement obtained with $Hybrid$ for large objects. Except for energy consumption, $120GBSSD$ has not presented significant results.

Figure 5(a) depicts $120GBSSD$ does not have a prominent IOPS/energy ratio, compared to other tecnologies. Regarding the ratio price/IO, Figure 5(b) indicates SSD has the highest cost.

E. Experiment V: Mixed

Table VII depicts the results for a workload composed of mixed operations, access patterns and distinct object sizes.

$120GBSSD$ has the smallest value for response time ($10.86ms$), influenced by small random requisitions ($4KB$ and rnd). $Hybrid$ has the highest IOPS (681.211) and $1TBHDD$ has the worst performance, except for energy consumption.

Figure 6(a) indicates $120GBSSD$ has the best energy efficiency, about 128.30% higher than $Hybrid$. However, Figure 6(b) shows $Hybrid$ has the best price-performance. Indeed, for the hybrid system, the high values for energy consumption and price are strongly compensated by system throughput.

F. Remarks

Hybrid systems generally have higher energy consumption. However, whenever performance requirements prevail over energy savings, hybrid storage is a prominent alternative, mainly for sequential accesses.

SSDs may have performance issues with sequential accesses, but they are suitable for services with small random requests. Concerning HDDs, results confirm the issues for processing small objects. Nevertheless, HDDs are still a feasible option for some systems represented by sequential accesses due to IOPS/energy and price/IOPS ratios.

VII. Conclusion

This paper presented an approach based on GSPN for performance and energy consumption assessment of storage systems. The proposed models allow the evaluation of different storage technologies and workloads. Experiments based on Storage Performance Council's guidelines illustrate the practical feasibility of our modeling approach for assisting system designers.

As future work, we are developing models for assessing the reliability and availability of hybrid storage systems.

Acknowledgment

This work has been supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq under grant 405224/2018-4 and 302373/2018-7. Besides, this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 88887.500766/2020-00.

References

[1] M. Zakarya, "Energy, performance and cost efficient datacenters: A survey," *Renewable and Sustainable Energy Reviews*, vol. 94, 2018.

[2] B. Mao, S. Wu, and H. Jiang, "Exploiting workload characteristics and service diversity to improve the availability of cloud storage systems," *Trans. on Parallel and Distributed Systems*, vol. 27, 2016.

[3] C. Karakoyunlu and J. A. Chandy, "Exploiting user metadata for energy-aware node allocation in a cloud storage system," *Journal of Computer and System Sciences*, vol. 82, 2016.

[4] R. H. Hariri, E. M. Fredericks, and K. M. Bowers, "Uncertainty in big data analytics: survey, opportunities, and challenges," *Journal of Big Data*, vol. 6, 2019.

[5] S. Singhal *et al.*, "A global survey on data deduplication," *International Journal of Grid and High Performance Computing*, vol. 10, 2018.

[6] S. Yin *et al.*, "Reed: A reliable energy-efficient raid," in *International Conference on Parallel Processing*. IEEE, 2015.

[7] J. Tai *et al.*, "Live data migration for reducing sla violations in multi-tiered storage systems," in *International Conference on Cloud Engineering*. IEEE, 2014.

[8] D. Boukhelef, J. Boukhobza, and K. Boukhalfa, "A cost model for dbaas storage," in *International Conference on Database and Expert Systems Applications*. Springer, 2016.

[9] A. Chikhaoui, K. Boukhalfa, and J. Boukhobza, "A cost model for hybrid storage systems in a cloud federations," in *Federated Conference on Computer Science and Information Systems*. IEEE, 2018.

[10] D. Lee, C. Min, and Y. I. Eom, "Effective flash-based ssd caching for high performance home cloud server," *Trans. on Consumer Electronics*, vol. 61, 2015.

[11] D. Boukhelef *et al.*, "Optimizing the cost of dbaas object placement in hybrid storage systems," *Future Generation Computer Systems*, vol. 93, 2019.

[12] B. Jiao *et al.*, "Duofs: An attempt at energy-saving and retaining reliability of storage systems," in *International Conference on Distributed Computing Systems*. IEEE, 2017.

[13] P. Maciel *et al.*, "Performance and dependability in service computing: Concepts, techniques and research directions, ser," *Premier Reference Source. Igi Global*, 2011.

[14] T. Kim and J. No, "Utilizing flash-memory ssd for developing hybrid filesystem," in *International Symposium on System Integration*. IEEE, 2014.

[15] W. Tan *et al.*, "Effectiveness assessment of solid-state drive used in big data services," in *Internation Conference on Web Service*. IEEE, 2014.

[16] M. Xie, L. Xia, and J. Xu, "State-dependent m/g/1/k queuing model for hard disk drives," in *Conference on Automation Science and Engineering*. IEEE, 2017.

[17] R. Salkhordeh, O. Mutlu, and H. Asadi, "An analytical model for performance and lifetime estimation of hybrid dram-nvm main memories," *Trans. on Computers*, 2019.

[18] J. Liu *et al.*, "Hybrid s-raid: A power-aware archival storage architecture," in *International Conference on Parallel and Distributed Computing, Applications and Technologies*. IEEE, 2012.

[19] G. Balbo, "Introduction to stochastic petri nets," in *Lectures on Formal Methods and Performance Analysis*. Springer, 2001.

[20] K. Trivedi, "Probability and statistics with reliability, queueing, and computer science applications, ed.: Jhon wiley &sons," 2002.

[21] A. A. Desrochers and R. Y. Al-Jaar, *Applications of Petri nets in manufacturing systems: modeling, control, and performance analysis*. IEEE, 1995.

[22] D. Meister and A. Brinkmann, "dedupv1: Improving deduplication throughput using solid state drives (ssd)," in *MSST*. IEEE, 2010.

[23] A. Valmari, "The state explosion problem," in *Lectures on Petri nets I: Basic models*. Springer, 1998.

[24] A. Melo *et al.*, "Dependability approach for evaluating software development risks," *IET Software*, vol. 9, 2015.

[25] W. W. Hsu and A. J. Smith, "The performance impact of i/o optimizations and disk improvements," *IBM Journal of Research and Development*, vol. 48, no. 2, pp. 255–289, 2004.

[26] W. Wu, W. Lin, C.-H. Hsu, and L. He, "Energy-efficient hadoop for big data analytics and computing: A systematic review and research insights," *Future Generation Computer Systems*, vol. 86, pp. 1351–1367, 2018.

[27] L. Mei, D. Feng, L. Zeng, J. Chen, and J. Liu, "Exploiting flash memory characteristics to improve performance of rais storage systems," *Frontiers of Computer Science*, vol. 13, no. 5, pp. 913–928, 2019.

[28] D. Richter, *Flash Memories*. Springer, 2016.

[29] M. Ajmone Marsan *et al.*, "A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems," *Trans. on Computer Systems*, vol. 2, 1984.

[30] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers*. John Wiley & Sons, 2010.

[31] S. P. Council, "Official Site," http://www.storageperformance.org, 2019, acessed: 2019/11/15.

[32] S. Yin *et al.*, "Duofs: A hybrid storage system balancing energy-efficiency, reliability, and performance," in *Euromicro International Conference on Parallel, Distributed and Network-based Processing*. IEEE, 2018.

[33] P. Saxena and P. Kumar, "Performance evaluation of hdd and ssd on 10gige, ipoib & rdma-ib with hadoop cluster performance benchmarking system," in *International Conference-Confluence The Next Generation Information Technology Summit*. IEEE, 2014.

[34] S. Park *et al.*, “A comprehensive study of energy efficiency and performance of flash-based ssd,” *Journal of Systems Architecture*, vol. 57, 2011.

[35] B. Montazeri *et al.*, “Homa: A receiver-driven low-latency transport protocol using network priorities,” in *Conference of the ACM Special Interest Group on Data Communication*. ACM, 2018.

[36] K. Nakashima, J. Kon, and S. Yamaguchi, “I/o performance improvement of secure big data analyses with application support on ssd cache,” in *International Conference on Ubiquitous Information Management and Communication*. ACM, 2018.

[37] Z. Li, M. Chen, A. Mukker, and E. Zadok, “On the trade-offs among performance, energy, and endurance in a versatile hybrid drive,” *ACM Trans. on Storage (TOS)*, vol. 11, no. 3, p. 13, 2015.

[38] Y. Kim, A. Gupta, B. Urgaonkar, P. Berman, and A. Sivasubramaniam, “Hybridplan: a capacity planning technique for projecting storage requirements in hybrid storage systems,” *The Journal of Supercomputing*, vol. 67, no. 1, pp. 277–303, 2014.

[39] B. Silva *et al.*, “Astro: An integrated environment for dependability and sustainability evaluation,” *Sustainable computing: informatics and systems*, vol. 3, 2013.

[40] A. Zimmermann *et al.*, “Towards version 4.0 of timenet,” in *MMB*. VDE, 2006.

[41] G. Bolch *et al.*, *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons, 2006.

[42] J. Wan *et al.*, “Deft-cache: A cost-effective and highly reliable ssd cache for raid storage,” in *International Parallel and Distributed Processing Symposium*. IEEE, 2017.

[43] L. Mei *et al.*, “A high-performance and high-reliability rais5 storage architecture with adaptive stripe,” in *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 2018.

[44] M. Lin *et al.*, “Efficient sequential data migration scheme considering dying data for hdd/ssd hybrid storage systems,” *IEEE Access*, 2017.